

CLAIMS

1. (currently amended) A media processing system interface comprising:

an input, coupled to a single instance of a source to access media content from the single instance of the source in response to requests for said media content, the media content having one or more subtypes; and

two or more outputs, each coupled to a media processing subsystem, wherein the interface routes at least a subset of ~~the received~~ media content received via the input to at least an individual one[s] of the two or more outputs based, at least in part, on ~~the~~ a media type of the subset.

2. (original) A media processing system interface according to claim 1, wherein the media processing system interface is a software object, exposed from an operating system on computing system implementing the media processing system.

3. (currently amended) A media processing system interface according to claim 1, wherein the interface receives requests from each of the one or more media processing subsystems, and initiates seeks into the single instance of the source for the requested content.

4. (currently amended) A media processing system interface according to claim 1, wherein the interface receives requests from each of the one or more media processing subsystems for particular ~~{source}-content[-]~~ from the single instance of the source and ignores the requests received from all but one of the media processing subsystems.

1 5. (currently amended) A media processing system interface
2 according to claim 1, wherein a source processing chain comprising each of the
3 media processing subsystems coupled through the interface to the single instance
4 of the source is removed from an active filter graph upon completion by each of
5 the media processing subsystems.

6 6. (currently amended) A media processing system interface
7 according to claim 5, wherein a first of the media processing subsystems instructs
8 a second of the media processing subsystems that it no longer requires content
9 from the single instance of the source, and the second media processing subsystem
10 informs a render engine controlling the filter graph to remove the source
11 processing chain when it no longer requires content from the single instance of the
12 source.

13 7. (currently amended) A media processing system interface
14 according to claim 6, wherein the render engine determines whether the source
15 processing chain may be required subsequently in this, or another media
16 processing project and, if so, caches the source processing chain for later retrieval
17 and integration in a processing project.

18 8. (currently amended) A media processing system interface
19 according to claim 5, wherein a first of the media processing subsystems request to
20 remove the source processing chain is ignored, wherein only the second of the
21 media processing subsystems can effectively request the source processing chain
22 to be removed from the active processing project.

23 9. (currently amended) A media processing system interface
24 according to claim 5, wherein a render engine controlling the project determines
25 whether at least a subset of the source filter chain will subsequently be required

1 and, if so, caches the source processing chain in local memory for subsequent
2 retrieval and integration into a processing project.

3 10. (original) A media processing system according to claim 1,
4 wherein the interface receives media content from a source and parses the received
5 content into its disparate content types.

6 11. (original) A media processing system according to claim 10,
7 wherein each of the media processing subsystems take one type of the parsed
8 media content for subsequent processing.

9 12. (original) A media processing system according to claim 10,
10 wherein the disparate content types include audio media content and video media
11 content.

12 13. (currently amended) A media processing system interface
13 according to claim 1, wherein the media processing subsystems include a
14 processing chain of software objects which manipulate the content retrieved from
15 single instance of the source in some fashion.

16 14. (original) A media processing system interface according to
17 claim 1, where the interface is a filter in a media processing filter graph.

18 15. (original) A media processing system interface according to
19 claim 1, the input is a software object, exposed from the operating system of a
20 computer implementing a media processing system, and implemented as an input
21 pin.

22 16. (original) A media processing system interface according to
23 claim 1, wherein each of the outputs are software objects, exposed from the
24 operating system of a computer implementing a media processing system, and
25 implemented as instances of an output pin.

17. (currently amended) A media processing system interface according to claim 1, wherein the interface is a parser object, implemented in a filter graph by render engine to enable multiple processing subsystems to access and receive content from ~~a~~the single instance of ~~a~~the source.

18. (original) A storage medium comprising a plurality of executable instructions which, when executed, implement a media processing system interface according to claim 1.

19. (original) A computer system comprising:
a storage medium having stored thereon a plurality of executable instructions; and
an execution unit, coupled to the storage medium, to execute at least a subset of the plurality of executable instructions to implement a media processing system interface according to claim 1.

20. (original) A filter graph implemented within a media processing system, the filter graph comprising:
a video processing subsystem to process video content;
an audio processing subsystem to process audio content; and
a parser object, coupling one or more of the video processing subsystem and the audio processing subsystem to a single instance of a multimedia source, to selectively provide the audio subsystem and video subsystem with requested audio content and video content, respectively.

21. (currently amended) A filter graph according to claim 20, wherein the parser is comprised of at least one input, couple to the single instance of the multimedia source, and two outputs, one each coupled to the video processing subsystem and the audio processing subsystem.

1 22. (original) A filter graph according to claim 20, wherein the
2 parser object receives request for content from each of audio processing subsystem
3 and the video processing subsystem and serializes such requests, processing them
4 in chronological order.

5 23. (original) A filter graph according to claim 20, wherein the
6 parser object receives request for content from each of the audio processing
7 subsystem and the video processing subsystem and ignores requests received on
8 all but a selected one of such audio processing subsystem or video processing
9 subsystem.

10 24. (original) A filter graph according to claim 23, wherein the video
11 processing subsystem is the selected one from which requests for content are acted
12 upon, while requests from the audio processing subsystem are ignored.

13 25. (currently amended) A filter graph according to claim 20,
14 wherein the parser object receives indications from one or more of the audio
15 processing subsystem and/or the video processing subsystem that the single
16 instance of the multimedia source is no longer required and, upon verifying that
17 neither processing subsystem requires further content from the single instance of
18 the multimedia source before a source filter chain including the parser object and
19 both media processing subsystems may be removed from the filter graph.

20 26. (original) A filter graph according to claim 25, wherein the
21 parser object informs a render engine that the source filter chain is no longer
22 required, whereupon the render engine may remove the source filter chain from
23 the filter graph.

24 27. (original) A filter graph according to claim 26, wherein the
25 render engine determines whether the source filter chain will be required

1 subsequently in this or another filter graph and, if so, caches the entire source filter
2 chain for later integration in the an appropriate filter graph.

3 28. (currently amended) A filter graph according to claim 20,
4 wherein the parser object only responds to an indication that the instance of the
5 multimedia source is no longer required from a select one of audio or video
6 [media] processing subsystems.

7 29. (original) A filter graph according to claim 28, wherein the select
8 media processing subsystem is the video processing subsystem.

9 30. (currently amended) A filter graph according to claim 29,
10 wherein the parser object ignores indications that the single instance of the
11 multimedia source is no longer required from the audio processing subsystem.

12 31. (currently amended) A filter graph according to claim 29,
13 wherein an element of the video processing subsystem checks with other media
14 processing subsystems coupled to the parser object to determine whether they, too,
15 no longer require content from the source before instructing the parser object that
16 the single instance of the multimedia source is no longer required.

17 32. (currently amended) A filter graph according to claim 29,
18 wherein the parser object ensures that both the video and audio processing
19 subsystems no longer require content from the single instance of the multimedia
20 source before unloading source filter strings comprising the audio processing
21 subsystem and the video processing subsystem.

22 33. (currently amended) A filter graph according to claim 20,
23 wherein the parser object receives indications from each of the audio and video
24 subsystems requesting content and resolves such requests based, at least in part, on
25 a priority of when such content is required in support of filter graph execution.

1 34. (original) A computing system comprising:
2 a storage medium having stored therein a plurality of executable
3 instructions; and

4 an execution unit, coupled to the storage medium, to execute at least a
5 subset of the plurality of executable instructions to implement a filter graph
6 according to claim 20.

7 35. (original) A storage medium comprising a plurality of executable
8 instructions which, when executed, implement a media processing system
9 including a parser object, to couple one or more media processing subsystems to a
10 single instance of a source to provide each of the coupled media processing
11 subsystems with requested content from the source.

12 36. (original) A storage medium according to claim 35, wherein each
13 of the media processing subsystems process media content of a particular type,
14 and wherein the parser object parses out that type of media content from the
15 source for provision to each of the media processing subsystems.

16 37. (original) A storage medium according to claim 35, wherein the
17 parser object receives multiple requests for content from the media processing
18 subsystems and serializes such requests, processing them in chronological order.

19 38. (original) A storage medium according to claim 35, wherein the
20 parser object receives requests for content from each of the one or more media
21 processing subsystems, but only acts on requests received from a single media
22 processing subsystem.

23 39. (original) A storage medium according to claim 35, wherein the
24 parser object receives indications from each of the one or more media processing
25

1 subsystems that content from the source is no longer required, but ignores all but
2 those indications received from a select media processing subsystem.

3 40. (original) A storage medium according to claim 35, wherein the
4 parser object receives an indication from one or a plurality of coupled media
5 processing subsystems that content from the source is no longer required, and
6 verifies that none of the remaining plurality of coupled media processing
7 subsystems require further content before releasing the source.

8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25